

NAME (as it appears on your UF ID): _____
(Please **PRINT**)
UF Student ID#: _____

----- CEN 4072/6070 Software Testing & Verification -----

Exam 1 -- Spring 2012

You have 90 minutes to work on this exam. It is a "closed-book/closed-notes" test. Pay attention to point values, since you may not have time to work all 25 problems. PRINT your name above NOW and sign the pledge at the bottom of the last page, if appropriate, when you are finished. PLEASE PRINT ANSWERS IN THE SPACE PROVIDED ONLY – PREFERABLY USING A BALLPOINT PEN TO INCREASE LEGIBILITY. Good luck!

1. (6 pts.) In their chapter, *"Making Meetings Work for Everyone,"* Gause and Weinberg offer several observations about, or recommendations for, making meetings more productive. Briefly describe the specific observation / recommendation illustrated by way of each of the two images below.

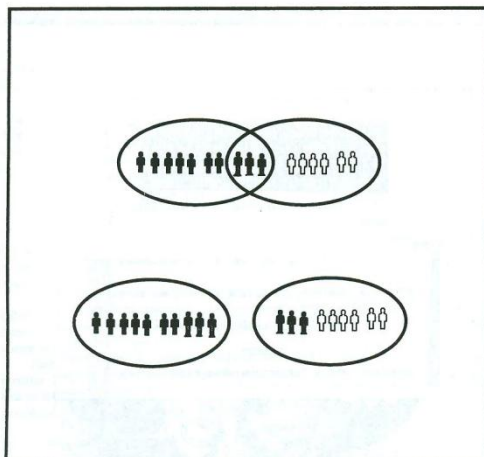
a.

observation/ recommendation:



b.

observation/recommendation:



2. (3 pts.) In *"The Psychology and Economics of Program Testing,"* Glenford Myers identifies what "in considering the issues of economics and psychology" he feels is "the most important single thing that one can learn about testing." Which one of the following is this single most important consideration? (Circle ONE only.)
- a. That testing is an extremely creative and intellectually challenging task
 - b. That test cases must be written for invalid and unexpected, as well as valid and expected, input conditions
 - c. That BOTH *data-driven* (i.e., black-box) and *logic-driven* (i.e., white-box) testing techniques must be employed
 - d. That one cannot test a program to guarantee that it is error free
 - e. (none of the above)
3. In discussing the limitations of "exhaustive path testing," Myers describes the issues of "missing paths" and "data-sensitivity errors."
- a. (2 pts.) What exactly does Myers mean by "**missing** paths"? That is, in what sense are the paths "missing" as opposed to just "non-existent"?
 - b. (4 pts.) Provide a simple, concrete example of a "data-sensitivity error" and clearly explain why exhaustive path testing might not detect it.
4. (2 pts.) Regression testing involves re-executing tests that have already been executed without revealing an error. Explain the purpose of this.

5. (3 pts.) Recall the question posed in class regarding the "Fisherman's Dilemma:" *You have 3 days for fishing and 2 lakes to choose from. Day 1 at Lake X nets 8 fish. Day 2 at Lake Y nets 32 fish. Which lake do you return to for day 3? Which one of the following is the primary conclusion drawn about testing in connection with this issue? (Circle ONE only.)*
- a. Like fishing, testing is an extremely creative and challenging task.
 - b. Testers should avoid attempting to test their own programs for the same reason that fishermen should avoid attempting to catch their own fish.
 - c. Tests should be designed and then executed to demonstrate the correspondence between an element and its specification.
 - d. Testers should execute programs with the *intent* of finding errors for the same reason that "real fishermen" should fish with the *intent* of catching fish.
 - e. Testing should never be undertaken under the tacit assumption that no errors will be found, just as fishing in a given lake should never be undertaken under the tacit assumption that no fish will be caught in that lake.
 - f. (none of the above)
6. (10 pts.) Use SIMPLE counter-examples to prove that Condition Coverage and Basis Paths Coverage are independent. EXPLAIN HOW YOUR COUNTER-EXAMPLES PROVE THIS. (Hint: consider utilizing a loop-free program with Cyclomatic Complexity = 3, number of paths = 4, and 2 primitive conditions in at least one of the two branch statement predicates.)

7. (6 pts.) In *"The clean-sheet approach to getting started,"* Edward Kit encourages the reader to identify "potential improvements you believe are key to renovating the software testing process in your organization." To start the process, he suggests three specific process improvement activities. What are they? (Note: If you cannot remember his suggestions, provide your own specific, concrete suggestions for process improvement that are directly related to three of his "six essentials of the software testing process.")
8. In "Design and code inspections to reduce errors in program development," Michael Fagan describes "a study of coding productivity" in which "a piece of the design of a large operating system component" was selected to study the effect of inspections at different levels on coding productivity. In describing the results of this study, Fagan notes that "an important aspect to consider in any production experiment involving human beings is the Hawthorne Effect."
- a. (3 pts.) What is the *Hawthorne Effect*?
- b. (3 pts.) What approach was used in Fagan's study to account for this effect?
9. (5 pts.) Identify the nominal **output** boundary values that, as discussed in class, should be explored for the `pow()` function based on the draft MAN page used in the black-box testing case study.

10. (3 pts.) Grady and Van Slack ("*Key Lessons In Achieving Widespread Inspection Use*") describe three major influences on software inspections at Hewlett-Packard: the history of hardware reviews at HP, Fagan's "very influential" 1976 paper, "*Design and Code Inspections to Reduce Errors in Program Development*," and the work of Tom Gilb, who "extended Fagan's inspection process in several important ways that represented a timely philosophical match with HP thinking." Which one of the following is identified by Grady and Van Slack as an important contribution by **Fagan**? (Circle ONE only.)

- a. recognizing that *all* software modifications are well worth inspecting
- b. presenting data that helped managers better visualize inspection results
- c. making training readily available to BOTH inspectors and managers.
- d. focusing inspections on EARLY life-cycle artifacts
- e. separating the tasks of defect collection and defect discrimination
- f. (none of the above)

11. (3 pts.) Some have suggested that Test-Driven Development (TDD) reflects an element of "methodological hyperbole." In particular, they argue the process model includes a step that serves no purpose other than to *dramatically reinforce* the principle that testable requirements for new functionality should be the focus BEFORE code is written to implement that functionality – a principle that, in their view, is already clearly reflected in the process requirement that a test be written and implemented before the new functionality is implemented. What is this additional, arguably "hyperbolic step" in the TDD process model?

12. (4 pts.) Suppose a program has three independent inputs, X, Y, and Z, with the following numbers of disjoint equivalence classes:

X: two classes Y: three classes Z: four classes

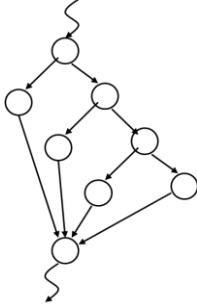
How many test cases would be required for "Strong Equivalence Class Testing (SECT)"? How many for "Weak Equivalence Class Testing (WECT)"?

Number of test cases required for SECT: _____

Number of test cases required for WECT: _____

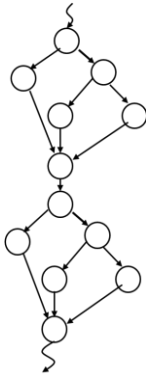
13. (16 pts.) For each of the following control flow graphs, indicate the number of complete program paths, test cases required for Basis Paths coverage, test cases required for Branch Coverage, and test cases required for Statement Coverage.

a.



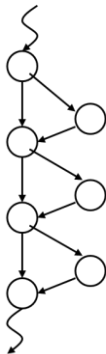
number of paths: _____
 test cases for Basis Paths Coverage: _____
 test cases for Branch Coverage: _____
 test cases for Statement Coverage: _____

b.



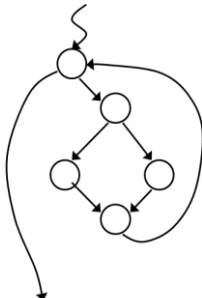
number of paths: _____
 test cases for Basis Paths Coverage: _____
 test cases for Branch Coverage: _____
 test cases for Statement Coverage: _____

c.



number of paths: _____
 test cases for Basis Paths Coverage: _____
 test cases for Branch Coverage: _____
 test cases for Statement Coverage: _____

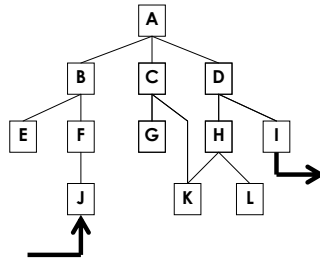
d.



number of paths: _____
 test cases for Basis Paths Coverage: _____
 test cases for Branch Coverage: _____
 test cases for Statement Coverage: _____

14. (3 pts.) Which one of the following best reflects the specific PURPOSE of the *Error Seeding approach* of Fault-Based Testing described in class? (Circle ONE only.)
- a. It is used to assess the error-revealing capability of a given test set with respect to a given program.
 - b. It is used to systematically track state mutations for the purpose of expressing path conditions in useful terms.
 - c. It is used to estimate the number of errors remaining in a program after a period of testing.
 - d. It is used to assess source- or object-code program compatibility with different operating environment versions that are identical except for some small change.
 - e. It is used to subvert encapsulation by mutating source code to allow inspection of private variables during white-box testing.
15. (5 pts.) In the context of discussing Usability Test, it was noted in class that "protocol analysis is used to identify usability bottlenecks." Briefly explain what "protocol analysis" involves and what is meant by a "usability bottleneck."
16. (4 pts.) What is an "invalid equivalence class" and why is it desirable to design test cases that cover such classes ONE AT A TIME?

17. Consider the module "calling hierarchy" below for a software component.



- a. (3 pts.) Identify the element(s) of "scaffolding" that may be required to test (only) modules C and K together during incremental integration testing, and briefly describe what role(s) each of these element(s) would play.

 - b. (3 pts.) Is "scaffolding" ever required outside the context of integration testing? Briefly explain your answer.
-
18. (9 pts.) It was noted in class that while the same basic pattern of testing procedural software also applies to testing object-oriented software, there are some feature of object-oriented languages that require special strategies and/or considerations. Identify 3 of the 4 object-oriented language features discussed in this context, and very briefly describe (using language along the lines of "testers must do..." or "testers may need...", etc.) the testing related strategy and/or consideration associated with each.

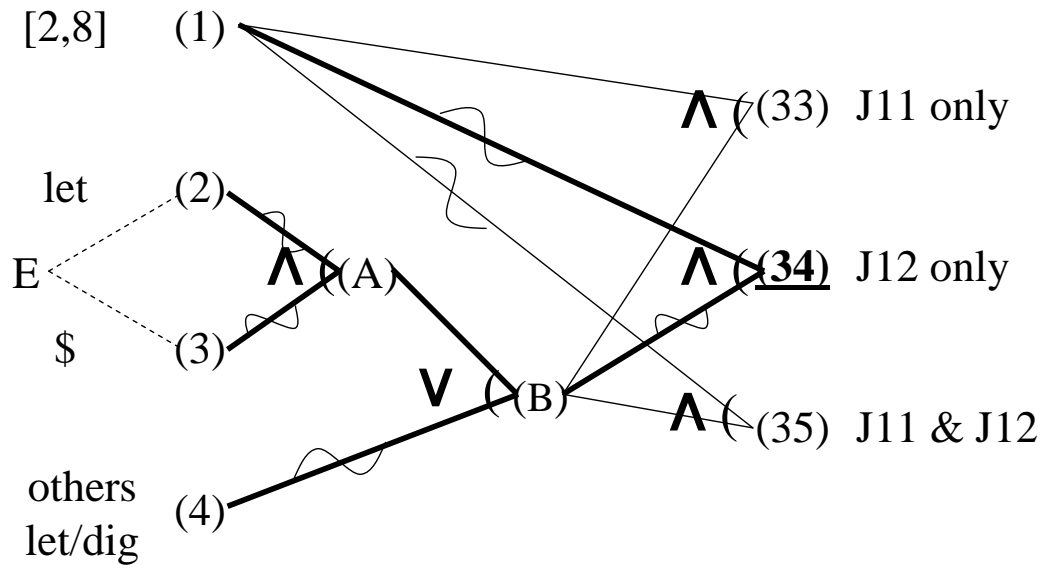
19. (3 pts.) During the discussion of testing tools, reference was made to the wonder and beauty of springtime in Paris...



Which one of the following testing tool types was this reference used to clandestinely introduce? (Circle ONE only.)

- a. Boundary value, equivalence class, and special value generators
 - b. Statistical test generators (for reliability, availability, and performance testing)
 - c. Test Harness Generators
 - d. Code Instrumentation tools
 - e. Data / Output Comparators
 - f. Keystroke Recorder and Playback Tools
 - g. (none of the above)
20. a. (3 pts.) In the context of Cause-Effect models with mutually exclusive Effects, what is the relationship between All Feasible Combinations of Cause Value ("AFCCV") coverage, and All Feasible Combinations of Connected Cause Values that Result in Each Effect being True ("Strategy #3") coverage? (Circle ONE only.)
- i. They are independent – neither subsumes the other
 - ii. "AFCCV" subsumes "Strategy #3"
 - iii. "Strategy #3" subsumes "AFCCV"
 - iv. They are equivalent – each subsumes the other
- b. (3 pts.) In the context of Cause-Effect models with NON-mutually exclusive Effects, what is the relationship between All Effects Covered with the Minimum number of test Cases ("AEMC") coverage, and All Feasible Combinations of Connected Cause Values that Result in Each Effect being True Plus Culling Rules ("Strategy #3 Plus Culling Rules") coverage? (Hint: consider the case where $(\text{Cause1} \vee \text{Cause2}) \Rightarrow \text{Effect1}$ and where $\text{Cause1} \Leftrightarrow \text{Cause2}$) (Circle ONE only.)
- i. They are independent – neither subsumes the other
 - ii. "AEMC" subsumes "Strategy #3 Plus Culling Rules"
 - iii. "Strategy #3 Plus Culling Rules" subsumes "AEMC"
 - iv. They are equivalent – each subsumes the other

21. (8 pts.) Recall the Cause-Effect graph depicting error conditions for the Symbol Table Storage specification:



Identify all feasible combinations of connected Cause values that result in Effect 34 being true. Enter a test case template for each combination identified in the test case coverage matrix below.

[illegible]

22. (11 pts.) Match each description below to the **SINGLE MOST APPROPRIATE TERM** among the following. (Note: terms may apply to none, one, or more than one description.)

- | | |
|------------------------|----------------------------------|
| A. Fault-based test | L. "Soak" test |
| B. Installability test | M. Device and configuration test |
| C. Thread test | N. Usability test |
| D. Alpha test | O. Serviceability test |
| E. Performance test | P. Beta test |
| F. Stress test | R. Security test |
| G. "Lights out" test | S. "Smoke" test |
| H. Regression test | T. Exhaustive test |
| I. Reliability test | U. Compatibility/conversion test |
| J. Post-test analysis | W. Causal analysis |
| K. Benchmarking | X. Test-driven development |

- ___ Typical coverage includes post-delivery change procedures (adaptive, perfective, and corrective scenarios), supporting documentation, and system diagnostic tools
- ___ Process aimed at identifying the origin of errors and approaches to eliminate future occurrences
- ___ End-user testing performed within the development environment prior to general release
- ___ Integrating program elements associated with a key program function
- ___ Automated, stand-alone testing not requiring human involvement
- ___ General practice of recording and comparing indices of performance, quality, cost, etc., to help identify "best practices"
- ___ Also known as "build verification"; initial test after a software build to detect catastrophic failure
- ___ Focus is on typical requirements that systems exhibit "graceful" failures and non-abrupt performance degradation.
- ___ Long term testing to detect latent problems such as memory leaks, performance problems, or other anomalies that may not be revealed by short term testing
- ___ May be automated by combining a keystroke recorder and playback tool with a data/output comparator
- ___ Appropriate interpretations for "failure" and "time" are critical.

23. a. (6 pts.) Provide a 5-node control-flow graph with appropriate node and edge annotations for the pseudocode program below that is suitable for dataflow coverage analysis. The 5 nodes of your graph should correspond to the line numbers given in the pseudocode.

```

1. input(X,Y)
   repeat
2.     Y := Y-X
       if X<=0 then
3.         X := X+1
4.     end_if_then
       until (Y<=0)
5. output(X,Y)

```

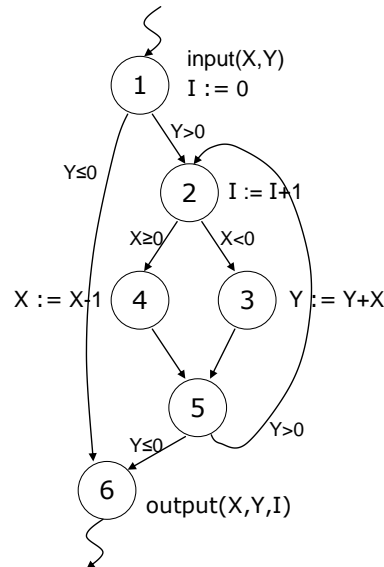
- b. (13 pts.) In the table below, list all definition-use pairs for variable X (only) and identify all (and only) the associated du-paths for each.

<u>du-pair</u>	<u>du-path(s)</u>

- c. (12 pts.) Give the **path condition** (in terms of the *initial* symbolic values of X,Y) for path <1,2,3,4,2,4,5>. Show ALL path condition predicates; do NOT simplify or combine terms.

- d. (2 pts.) For what initial values of X and Y would this path be sensitized?

24. (3 pts.) Recall the control-flow graph given in the Problem Set 3 Solution Notes for analyzing the dataflow coverage of a program with a while loop containing a nested if-then-else statement:



It was claimed that test case set 2, comprised of two test cases covering execution paths $\langle 1, 2, 3, 5, 6 \rangle$ and $\langle 1, 2, 4, 5, 2, 4, 5, 2, 4, 5, 2, 3, 5, 6 \rangle$, met the criteria for Branch Coverage, even though edge $\langle 1, 6 \rangle$ is not traversed. Was this an error in the Solution Notes? Briefly explain your answer.

25. (2 pts.) As discussed in class, software reliability may be measured using *statistical testing* based on an operational profile. What is an "operational profile"?

On my honor, I have neither given nor received unauthorized aid on this exam and I pledge not to divulge information regarding its contents to those who have not yet taken it.

SIGNATURE